IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 January 2024, accepted 7 May 2024, date of publication 23 May 2024, date of current version 31 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3401733

RESEARCH ARTICLE

FL-ToLeD: An Improved Lightweight Attention Convolutional Neural Network Model for Tomato Leaf Diseases Classification for Low-End Devices

MAHMOUD H. ALNAMOLY[1], ANAR A. HADY[2,3], (Member, IEEE), SHERINE M. ABD EL-KADER[3], AND IBRAHIM EL-HENAWY[1]

[1]Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt
[2]Faculty of Informatics and Computer Sciences, The British University in Egypt (BUE), El Shorouk City, Cairo 11837, Egypt
[3]Computers and Systems Department, Electronics Research Institute, Cairo 12611, Egypt

Corresponding author: Anar A. Hady (anar.abdelhady@bue.edu.eg)

ABSTRACT The agricultural sector is still a major provider of many countries' economies, but diseases that continuously infect plants represent continuous threats to agriculture and cause massive losses to the country's economy. In this study, a faster and lightweight tomato leaves diseases detection model was proposed for tomato disease classification based on a soft attention mechanism with a depth-wise separable convolution layer. With a model size of 2.5 MB and 221,594 trainable parameters, the proposed model achieved 99.5%, 99.10%, 99.04% for training, validation and testing accuracy respectively, and 99 % for each of precision, recall, and f1-score, it also achieved 99.90% for ROC-AUC with average inference time of 2.06924 μs. The proposed model outperformed Ulutaş and Aslantaş (2023) by 2.2% in terms of accuracy, precision, recall and f1-score. Additionally, it performed better than Agarwal (2023), Abbas (2021), and Verma (2020) in terms of accuracy, precision, recall, and f1-score by 8%, 2%, and 6%, respectively. It also outperformed Arshad (2023) by 4.77%, 8.92%, 35.18% and 5.11% in terms of accuracy, precision, recall and f1-score, respectively. Furthermore, the proposed model is 90 times smaller than Verma (2020) and 2.5 times smaller than Ulutaş and Aslantaş (2023) in terms of model size. All this makes the proposed model more suitable for low-end devices in precision agriculture.

INDEX TERMS Convolutional neural networks, soft attention, deep learning, tomato, classification, precision agriculture.

I. INTRODUCTION

Tomato or Solanum lycopersicum as they are called, is one of the most popular vegetables and one of the cash crops that is grown all over the world. Tomatoes in Egypt are considered one of the most important food sources; Egypt is the fifth largest producer around the world, with an annual production of 6.62 million metric tons [1]. However, its production is deteriorating owing to the exposure of the crop to various diseases such as early blight, late blight, leaf mold and other diseases. For example, early blight is responsible for significant yield losses worldwide and accounts for up to 78% of the yield declines [2]. So, in order to meet this urgent need, it is crucial to enhance agricultural yields and safeguard crops [3] by continuously monitoring plants as it is very important to detect plant diseases at their earlier stages to avoid huge losses in our production [4]. The conventional method to detect the plant disease requires manual examination of infected leaves by visual cues or chemical analysis of affected areas to note the differences on the leaf such as

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Li.

brown or black holes, which is performed under low detection efficiency and with high error as it requires good professional knowledge [5].

Owing to the continuous development of artificial intelligence techniques over the past few years, the process of automatic plant disease detection has become more flexible and is commonly used in smart farming. These AI systems do not require human intervention and are fast, inexpensive, and more accurate than traditional techniques and are currently applied to many agricultural applications on Smart farming [6]. Recently, one of the mostly used deep learning algorithms for classification and recognition tasks is Convolutional Neural Network (CNN) algorithm which can avoid complex preprocessing of traditional machine learning algorithms by automatically extracting features directly from the input images without our supervision [7], [8]. AI systems used in precision agriculture usually require a small model size that recognizes diseases in the least possible time for low computational power devices with small memory but this can lead to non-satisfactory inference accuracy. Therefore, in this research article, we seek to achieve the trade-off between model accuracy and complexity by developing a customized CNN model called FL-ToLeD which stands for faster and lightweight tomato leaves diseases detection.

Whereas the main contributions are as follows:

1. A faster and lightweight convolutional neural network model called FL-ToLeD is developed for tomato disease classification.
2. Acheiving the trade-off between model performance and computational complexity which makes it a more suitable choice for low-end devices.
3. Using a soft attention-based mechanism with a test accuracy of 99.04%, making it more suitable in case of early stages of disease with small lesions.
4. Using only 0.221 million of trainable parameters with a model size=2.5 MB, through using depth-wise separable convolution layers with batch normalization layers.
5. Performing a study to evaluate the reliability of the FL-ToLeD model and the individual impact of each component within it. Qualitative analysis and quantitative analysis were performed for the FL-ToLeD model.
6. Outperforming most of the existing tomato disease classification' models across seven different evaluation criteria.

The rest of this paper is organized as follows. Section II provides a summary of related work. Section III elaborates the proposed convolutional neural network model for tomato plant disease classification. The experimental results of the proposed model are presented in Section IV. Section V provides a summary of the original contributions, limitation and discusses future work.

II. RELATED WORK

Ahmed et.al [9] proposed a transfer learning-based strategy to identify diseases in tomato leaves using PlantVillage dataset. This method's initial step involves enhancing the leaf images with illumination correction using a powerful preprocessing technique in order to improve classification. The illumination issue that persisted in the dataset has been resolved with the use of the adaptive contrast enhancement technique. Following that, a hybrid model made up of a classifier network and a pre-trained MobileNetV2 architecture was used for feature extraction. To prevent data leakage and address the problem of class imbalance, they applied a runtime augmentation instead of applying the traditional data augmentation. Their model achieved an accuracy of 99.30% over 2.4 million training parameters with a model size about 9.60MB. The main drawbacks of using transfer learning are overfitting and negative transfer.

Amreen et.al [10] suggested a deep learning approach for tomato plant disease detection. They created synthetic images using Conditional Generative Adversarial Network (C-GAN) as a data augmentation action to increase the size of their dataset and to prevent their model from the problem of overfitting. They applied a transfer learning model called DenseNet121 to classify their dataset which consisted of ten different types of real and synthetic tomato disease images. The accuracy of this model, which can be divided into five classes, seven classes, and ten classes, was tested on the PlantVillage dataset and was 99.51%, 98.65%, and 97.11%, respectively. The over-fitting issue is avoided by the suggested data augmentation technique (C-GAN), which also increases the generalizability of the network. There are 1,735,904 trainable parameters in this model. The accuracy is good but not the best one compared to other models.

Trivedi et al. [11] proposed a CNN classification model for tomato diseases. The images were subjected to some preprocessing, as well as image segmentation. Secondly, varying hyper-parameters of the CNN model were applied for processing the images in a further way. Finally, the proposed model extracted additional characteristics from the images, such as texture, color, and edges, etc., which will then be classified. The suggested model performed with a 98.49% accuracy utilizing 1,422,542 training parameters after being tested and trained on a dataset of 3000 photos of tomato leaves. It is not a good choice for embedded low-end devices as it takes up a large size of the disk space. Alhaj et al. [12] employed transfer learning to identify tomato leaf disease using the InceptionV3 model. It said that transfer learning cuts down on execution time and that it obtained a 99.8% accuracy rate, but it made no mention to time, complexity or the amount of training parameters that were used the model was made available as a cloud-based web application. The main drawbacks of using transfer learning are overfitting and negative transfer.

Khamparia et al. [13] proposed a hybrid model that consisted of a convolutional Auto-encoder and convolutional neural network to detect the disease on (tomatoes, potatoes, and maize). They used convolutional auto-encoder for reducing the number of features on the final features map

to reduce the number of trainable parameters. Their model achieved a training accuracy of about 100% but on the other hand, the testing accuracy was about 86.78% in which their model was over-fitted on the training data. They trained their model on a dataset consisting of 900 images divided into 6 classes and finally, they said that their model had a 97.50% accuracy rate and used about 3.3 million training parameters.

Agarwal et al. [14] developed a convolutional neural network model which consisted of three convolutional layers, three max pooling layers and two fully connected layers to identify the disease of tomato plants. Before training their model, they used data augmentation techniques to balance the data in each class as the class's images aren't balanced. After that, they trained and tested their model using 17500 images from PlantVillage dataset which was divided into 10000 images for training, 7000 for validation and 500 for testing. Their model achieved a testing accuracy of 91.2% with a model size of about 1.5 MB and about 208802 training parameters. The problem of this model was the low accuracy compared to other models.

Verma et al. [15] employed a three well known convolutional neural networks i.e. SqueezeNet, Inception V3 and AlexNet for the purpose of evaluating the severity of the late blight disease on tomato plants using feature extraction and transfer learning techniques. For evaluating the severity of the disease, they separated the images into the PlantVillage dataset based on their disease stage. They selected 355 colored images in their early stage, 347 colored images in their middle stage and 382 colored images in their end stage. They applied the model based on only transfer learning and by using a multiclass support vector machine as a classifier algorithm after extracting the features from images. AlexNet outperformed the other two networks in both approaches, with accuracy rates of 89.69% and 93.4%, respectively. Their model used about 61 million training parameters and the size of the model is about 227MB.The drawback of this model was the low accuracy compared to other models, high computational complexity and it takes up a large size of disk space which is not suitable for low end devices.

Karthik et al. [16] proposed two deep learning approaches for detecting the type of three diseases on the tomato leaves plants namely leaf mold, late blight, and early blight. For better classification they needed to learn or extract the significant features so that they applied residual learning CNN. To improve the accuracy of residual CNN and specifically learn significant feature maps, they employed an attention mechanism on top of it as the attention mechanism to give more weightage to the significant features for accurate classification. This was the first attempt to implement an attention based residual CNN. Their model was trained on 95999 augmented images from the PlantVillage dataset after applying a central zoom and random crop & zoom techniques to focus only on the leaf and not the background information. Their model achieved an accuracy of 98% on the validation sets in the 5-fold cross-validation and used around 600K training parameters.

Elhassouny et al. [17] developed a smart mobile application embedded with a deep convolutional neural network based on a well-known model called MobileNet to identify the most ten common types of tomato leaf diseases. The model accepted a colored images of size 224 $\times$ 224 as an input and it consisted of a set of 3 $\times$ 3 Depth-wise separable convolutions layers for reducing the number of computations followed by batch normalization and ReLU activation function and ended by average Pooling layer, dense layer with SoftMax function for 10 diseases classes. They trained their model on a PlantVillage dataset consisting of 7176 images of tomato leaves and achieved an accuracy of 90.3%. This model also suffered of low accuracy compared to other models.

Costa et al. [18] proposed a modified deep learning model based on well-known InceptionV3 and CNN using a hierarchical approach for the purpose of identifying 16 types of diseases in tomatoes, apples, and peaches. They trained their model on a PlantVillage dataset consisting of 24,000 images of tomato, apple, and peache leaves and achieved an accuracy of 97.74%.

III. PROPOSED MODEL

Model performance and computational complexity are always trade-offs because if detection performance is assured, the model must fully understand the features of the image, which increases the computational complexity and slows down the detection speed [19]. Therefore, in our work, we designed a lightweight convolutional neural network model called FL-ToLeD for tomato disease classification based on a soft attention mechanism with depth-wise separable convolution layers. FL-ToLeD achieves a good performance with minimum trainable parameters and low computational complexity and thus will be more suitable for low end devices.

FL-ToLeD consists of four blocks for feature extraction as shown in the conceptual pipeline of the FL-ToLeD model in Figure 1; the first two blocks consist of standard convolutional layers with a pooling layer, followed by a dropout layer of a rate 0.15%. The third block consists of two depth-wise separable convolutions layers followed by a batch normalization layer, the last block of the feature extraction phase is a soft attention-based mechanism. These four blocks will be discussed in the following sections in details.

Instead of using standard fully connected layers in the classification phase, we used a globalaveragepooling2D layer [20] as it can reduce the amount of trainable parameters. After that, a SoftMax layer [16] with ten classes is applied to assign decimal probabilities to each class.

Finally, a mathematical function called argmax [9] is applied to return the index of maximum value among the ten values to be the predicted class label. FL-ToLeD applied each of data augmentation and data normalization techniques [12], [21]. We applied runtime image augmentation to expand the

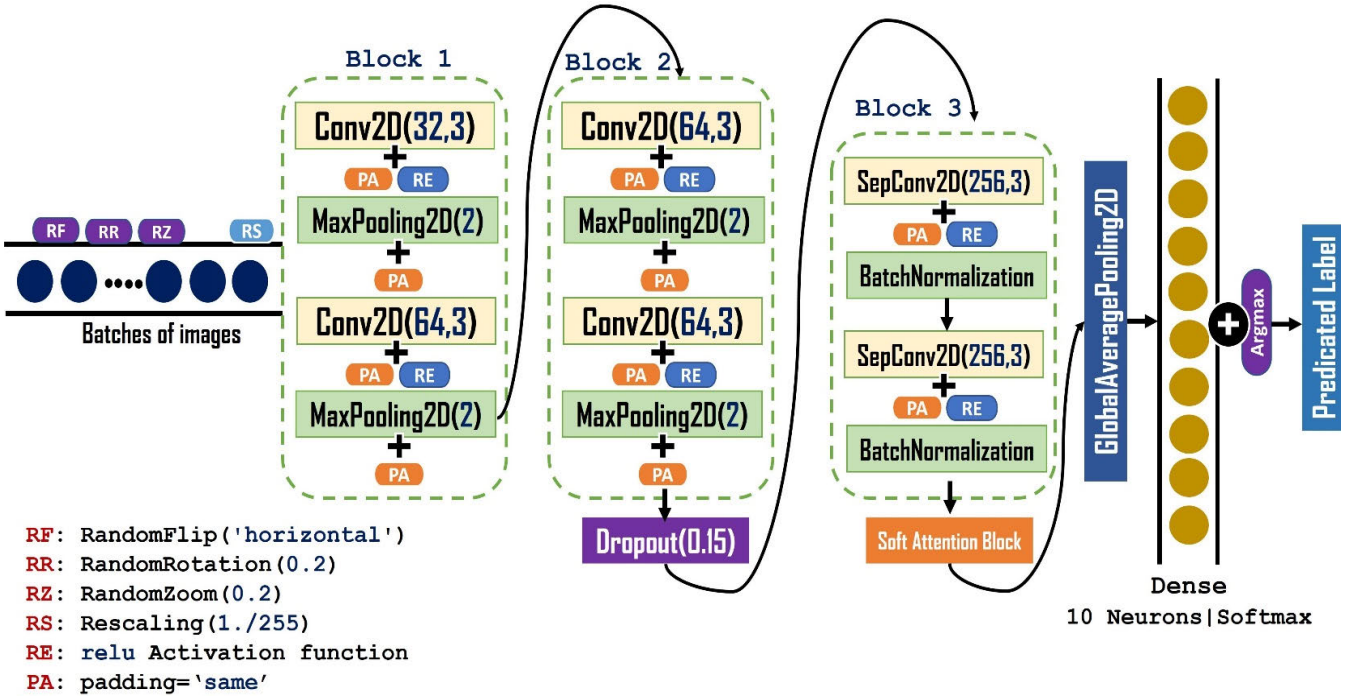

FIGURE 1. The conceptual pipeline of the FL-ToLeD model.

training dataset used such as random-flip, random-rotation, random-zoom with an input image size of (224 × 224×3). We also normalize inputs by changing the range of pixel intensities to bring values into a common range, and it is typically implemented by scaling all pixel values to range between 0 and 1. This is done by dividing all pixel values in each image by 255.

A. STANDARD CONVOLUTIONAL LAYERS (BLOCK 1 & BLOCK 2)

Three dimensions of each image - two spatial (width and height) and one channel - are used to extract features during a standard convolution operation. Therefore, the convolutional kernel must simultaneously describe cross-channel correlations and spatial relations as shown in (1). FL-ToLeD's Block 1 consists of two convolutional layers [22] with two pooling layers [23], the first convolutional layer applies learnable filters on the images with a size of 32 × 32, After that a 'same' padding operation and Relu activation function have been applied. These learnable filters [24] are a smaller matrices (K) that moves through the input image (x) from left to right from top to bottom and calculates the dot product of the filter weight (w) and the corresponding input image patch. Relu activation function [25] is one of the mostly used activation functions for convolution layers which is applied after each convolution operation to add some nonlinearity to our model, this is done by converting any negative values into the extracted feature map. The second convolutional layer uses a filter of size 64×64. Block 2 consists of the same layers of block 1 with different filter size on the first convolutional layer. To limit the amount of parameters the network must employ by reducing the spatial size of the convolutional output; FL-ToLeD applied a max pooling layer [23] with a size of 2 × 2 after each convolutional layer as a pooling layer.

$$Conv(W, x)_{(i,j)} = \sum_{m,n,k}^{M,N,K} w_{(m,n,k)} \cdot x_{(i+m,j+n,k)} \qquad (1)$$

where, W is the trainable weight matrix of the convolutional kernels, (i, j) is the coordinate point of the output feature maps, and x is the input feature map of the convolutional layer. The three convolutional kernel's dimensions are (m, n, k).

B. DEPTH-WISE SEPARABLE CONVOLUTIONAL LAYERS WITH BATCH NORMALIZATION (BLOCK 3)

Block 3 of FL-ToLeD consists of depth-wise separable convolution layers followed by a 'same' padding operation and Relu activation function which has a great impact on reducing the number of trainable parameters and computational complexity and batch normalization layers after each depth-wise separable convolution operation for improving the model performance.

On the above section, we mentioned that during the standard convolution operation, the three dimensions of each image, which are one channel dimension and two spatial dimensions (height and width), are used to extract features. But in the case of depth-wise separable convolution operations, the spatial correlations and cross-channel correlations can be extracted separately as shown in Figure 2.

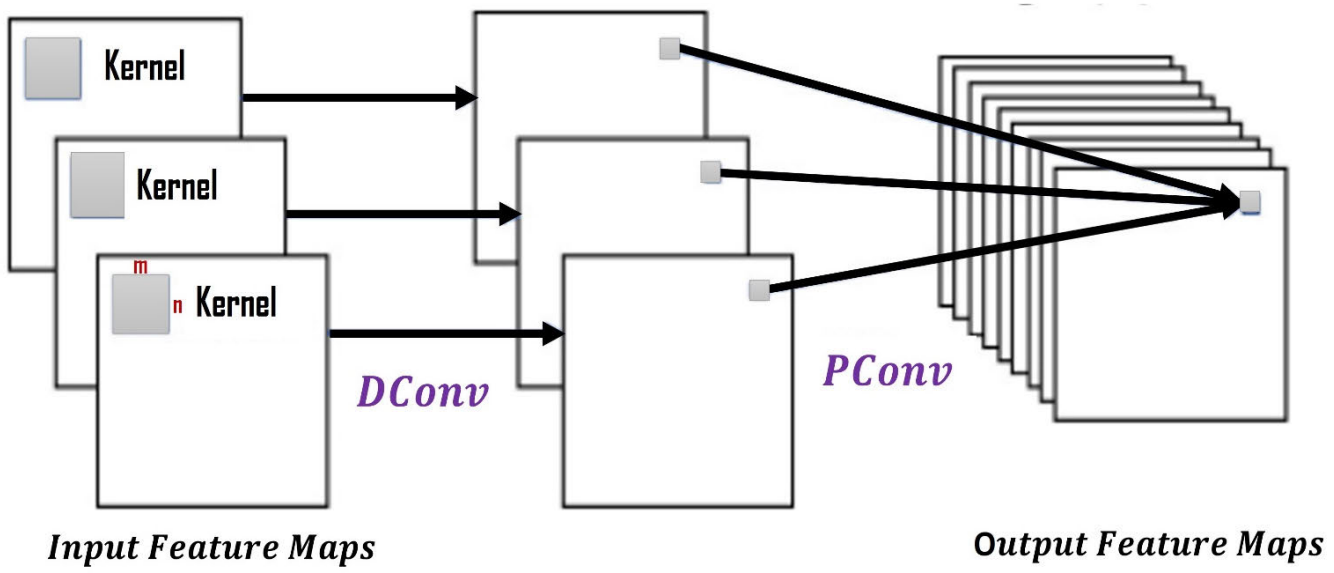

FIGURE 2. Two stage of depth-wise separable convolutions operations.

A depth-wise convolution is the initial stage of a depth-wise separable convolution. Each input channel receives only one filter, allowing each channel to output one feature map. The number of channels remain the same after depth-wise convolution operation [26].

There are two stages of depth-wise separable convolution; depth-wise convolution (filtering stage) as shown in (2) and point-wise convolution (combination stage) as shown in (3)

$$\boldsymbol{DConv}(\boldsymbol{W},\boldsymbol{x})_{(\boldsymbol{i},\boldsymbol{j})} = \sum_{m,n}^{M,N} \boldsymbol{w}_{(\boldsymbol{m},\boldsymbol{n})} \cdot \boldsymbol{x}_{(\boldsymbol{i}+\boldsymbol{m},\boldsymbol{j}+\boldsymbol{n})} \tag{2}$$

where, $\boldsymbol{W}$ is the trainable weight matrix of the convolutional kernels, $(\boldsymbol{i},\boldsymbol{j})$ is the coordinate point of the output feature maps, and $\boldsymbol{x}$ is the input feature map of the convolutional layer. The convolutional kernel's spatial dimensions (height and width) are ($\boldsymbol{m}$ ***and*** $\boldsymbol{n}$).

The outputs of the depth convolution operation are combined in the second stage using a 1×1 convolution (also known as a point convolution). In order to extract spatial features, pointwise convolution operation is performed. Although the channel number can changed, the spatial size of feature maps is not altered [27].

$$PConv\,(W,x)_{(i,j)} = \sum_{k}^{K} W_k \cdot x_{(i,j)} \tag{3}$$

where, **W** is the trainable weight matrix of the convolutional kernels, **(i, j)** is the coordinate point of the output feature maps, and $\boldsymbol{x}$ is the input feature map of the convolutional layer. The convolutional kernel's spatial dimensions (height and width) are **m** and **n** for all feature maps **k**.

In image classification, depth-wise separable convolution operations have been shown to be effective; It can avoid extracting some redundant features and greatly reduce the necessary ones, as by using these two depth-wise separable convolution operations, the complexity of the model and its parameters are significantly reduced compared to the standard convolutional layers.

In case of standard convolutional layers according to (1), the total required parameters are (m × n × k × o) if the number of output channel is o, while the total required parameters in case of depth-wise separable convolutional layers according to formula (2,3) are (m × n × k + k × o).

Assume that we have an input feature map of size 10 × 10×3 and the desired output tensor is of size 10 × 10×256 with filter of size 7 × 7×3. If we use the normal 2D Convolutions, then the number of multiplications required is (10 × 10) x (7 × 7×3) x (256) = **3,763,200**, but in our case we split into single channels, so 7 × 7×1 filter is required in place of 7 × 7×3, and since there are three channels, so the total number of 7 × 7×1 filters required is 3, so that we need (10×10) x (7×7×1) x (3) = **14,700** for filtering and (10×10) x (1 × 1×3) x (256) = **76,800** for combining as the total number of feature maps k required is 256 so that the total number of multiplications = **14,700** + **76,800** = **91,500**.

So, a 2D convolution will require 3,763,200 multiplications, while a Depth-wise Separable convolution will require only 91,500 multiplications to reach the same output. Finally, 3,763,200 / 91,500 = **41x** less multiplications are required. Since the depth separable convolution has a lower trainable parameter number than a typical CNN, the risk of overfitting on small datasets is minimized [28].

Moreover, the time complexity of standard convolutional layers applied in block 1 and block 2 is $\sim \boldsymbol{O}(\boldsymbol{M^2} * \boldsymbol{K^2} * \boldsymbol{C_{in}} * \boldsymbol{C_{out}})$ while the time complexity of the depth-wise separable convolution layer applied in block 3 is

Algorithm 1 Soft-Attention Mechanism

Input: Feature maps $\boldsymbol{x_{Inp}}$ from the Depth-wise Separable Convolutions block.

1. Applying max-pooling operation with kernel size = 2 on input feature map $\boldsymbol{x_{Inp}}$

$$\boldsymbol{x_{MP}} = \text{MaxPooling2D}(\boldsymbol{x_{Inp}})$$

2. Applying a 3D convolution operation with kernels or weights $\boldsymbol{W_k}$ where, $\boldsymbol{k}$= **16** is the number of 3D weights on input feature map $\boldsymbol{x_{Inp}}$.
3. The output of this convolution operation is normalized using **SoftMax** function to generate K-attention maps. **SoftMax** is commonly used to transform the raw attention scores obtained from the dot product into a valid probability distribution, with all values between 0 and 1 ensuring that the sum of the attention weights equals 1. This normalization allows the model to effectively focus on certain parts on the input image.

$$K - attentionmaps = \sum_{k=1}^{K} \text{softmax}(\boldsymbol{W_k} * \boldsymbol{x_{Inp}})$$

4. These k-attention maps (softmax scores) are aggregated to produce a unified attention map that acts as a weighting function α.

$$\boldsymbol{\alpha} = \oplus(\text{K-attention maps})$$

5. This α is then multiplied with input feature map $\boldsymbol{x_{Inp}}$ to attentively scale the salient feature values, which is further scaled by a learnable scalar $\boldsymbol{\gamma}$ = 0.01. This scaling process helps the model learn faster.

$$\boldsymbol{f_{sa}} = \boldsymbol{\gamma}\ (\boldsymbol{\alpha} * \boldsymbol{x_{Inp}})$$

6. These scaled features $\boldsymbol{f_{sa}}$ are concatenated with the input feature map $\boldsymbol{x_{Inp}}$ to produce a new feature map $\boldsymbol{x'_{Inp}}$
7. Applying a max-pooling operation with kernel size=2 on this new feature map $\boldsymbol{x'_{Inp}}$ to down sample the feature map.

$$\boldsymbol{x'_{MP}} = \text{MaxPooling2D}(\boldsymbol{x'_{Inp}})$$

8. Concatenate this feature map $\boldsymbol{x'_{MP}}$ with $\boldsymbol{x_{MP}}$ that was generated on step 1 to create the final feature map $\boldsymbol{x_{Con}}$
9. Applying a Relu activation function on $\boldsymbol{x_{Con}}$ to set any negative value to zero to produce the final attention feature map $\boldsymbol{x_{attention}}$
10. Passing $\boldsymbol{x_{attention}}$ to the globalaveragepooling2D layer to start the classification process.

Output: Attention Feature map $\boldsymbol{x_{attention}}$ passed forward to the classification layer.

$\sim \boldsymbol{O(M^2 * K^2 * C_{in} + M^2 * C_{in} * C_{out})}$ where M is the size of feature map, K is the size of kernel, C_{in} is the number of input channels, C_{out} is the number of output channels.

One the other side, to make the training of FL-ToLeD faster and more stable we use an algorithmic technique called batch normalization [29] which is applied after each depth-wise separable convolution operation. However, batch normalization increases the computational cost and memory usage, it improves the performance of the model. Finally, the combination of depth-wise separable convolution layer with batch normalization layers has a high significance on reducing computational complexity and increasing the performance of the model.

C. SOFT ATTENTION-BASED MECHANISM BLOCK

Attention mechanisms are a layer of neural networks added to deep learning models to focus their attention on specific pieces of data based on different weights assigned to different parts. It enhances deep learning models by selectively focusing on important input elements, improving prediction performance and computational efficiency. It prioritizes and emphasizes relevant information, and serves as a highlight to improve the overall performance of the model.

Soft attention mechanism is one of the important classes of attention mechanism, and it has been used in many fields of computer vision, such as classification, segmentation, detection, video processing, etc. The primary goal of Soft-Attention is to enhance the value of important features and suppress noisy features.

Lesions of small size on the leaves are not visible in the early stages of the disease. Additionally, it is quite challenging to extract the lesions features. However, the important patterns for differentiating several types of tomato disease are these very subtle differences in color and texture of the lesions. So, to extract and focus more on these lesions' features accurately, soft attention is implemented.

Instead of processing the entire image equally, Soft Attention-based mechanisms [30], [31] highlight or pay attention to a specific area of interest, increasing the value of relevant features and suppressing distracting features. By multiplying input feature maps with low weights, it invalidates the irrelevant regions of the image. As a result, the low attention regions have weights that are closer to zero. With more focused features, the model performance is enhanced.

In the attention-based mechanism, each feature in the image is assigned an attention score. Higher attention scores indicate greater importance, while lower scores indicate less importance.

The procedure of soft attention-based mechanism with the contribution of the individual components on it is discussed in details in Algorithm 1. FL-ToLeD applies soft attention-based mechanism on the feature maps produced from Depth-wise Separable Convolutions block as shown in Figure 3.

Last but not least, the soft attention method will improve the performance of our model, especially in early disease cases, and it will also reduce the bias caused by noise present

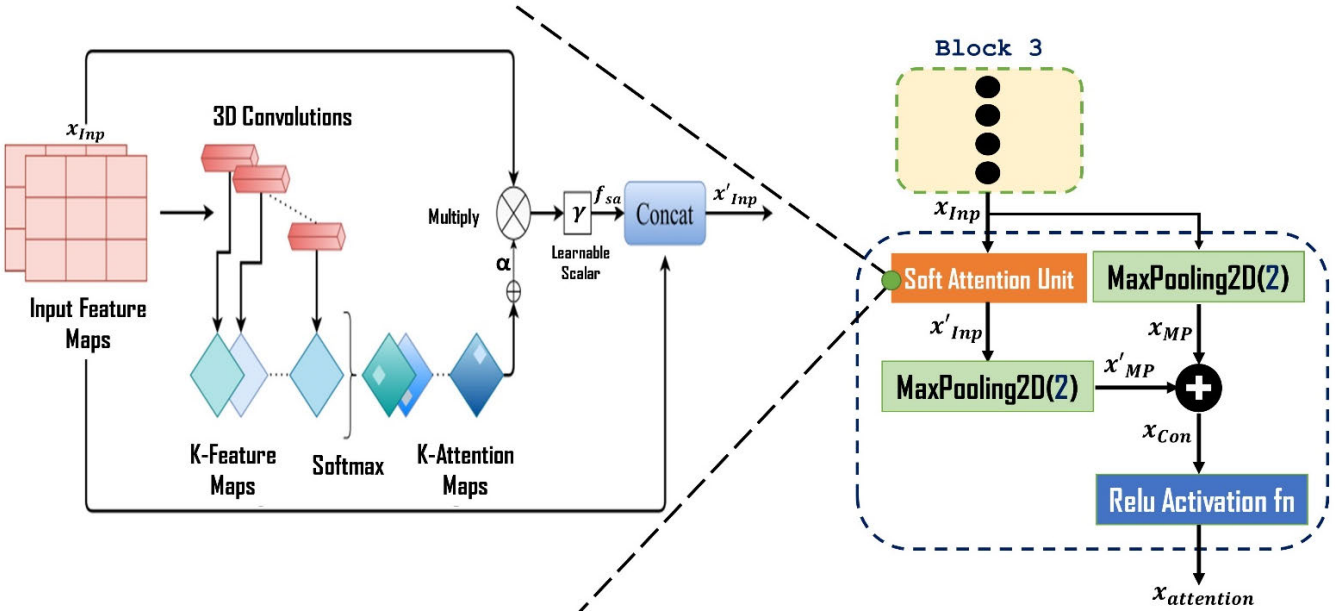

FIGURE 3. Soft attention-based mechanism operations.

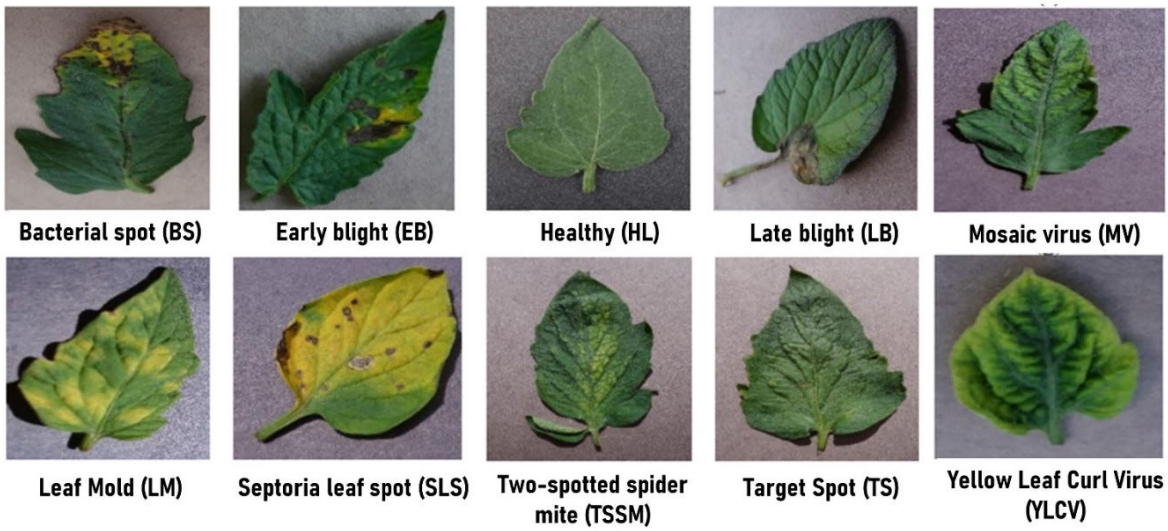

FIGURE 4. The ten classes of the tomato leaf diseases.

in the leaf image as it helps suppress noisy features. We'll go into more depth about all of this later.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. DATASET

Tomato leaf disease images from the Plant Village dataset [32] are used to train and test FL-ToLeD. The PlantVillage dataset is the most comprehensive, open-access collection of plant leaf imagery used for disease diagnosis comprising 54,309 images of healthy and diseased leaves belonging to 14 crops, which have been categorized by plant pathologists as tomato, potato, apples, soybeans, grapes, corn, etc. We extracted tomato leaf images from it as our target crop which was divided into one healthy category and nine disease classes: 1) Bacterial spot (BS), 2) Early blight (EB), 3) Late blight (LB), 4) Leaf Mold (LM), 5) Septoria leaf spot (SLS), 6) Target Spot (TS), 7) Spider mites: Two-spotted spider mite (TSSM), 8) Yellow Leaf Curl Virus (YLCV), 9) Mosaic virus (MV) as shown in Figure 4. The potential sources of bias on Plant-Village are; showing an unbalanced nature in terms of the number of images available for different plant diseases [33]. These unbalanced data can significantly impact any model performance in which the model in this case, focuses on the accuracy of the dominant

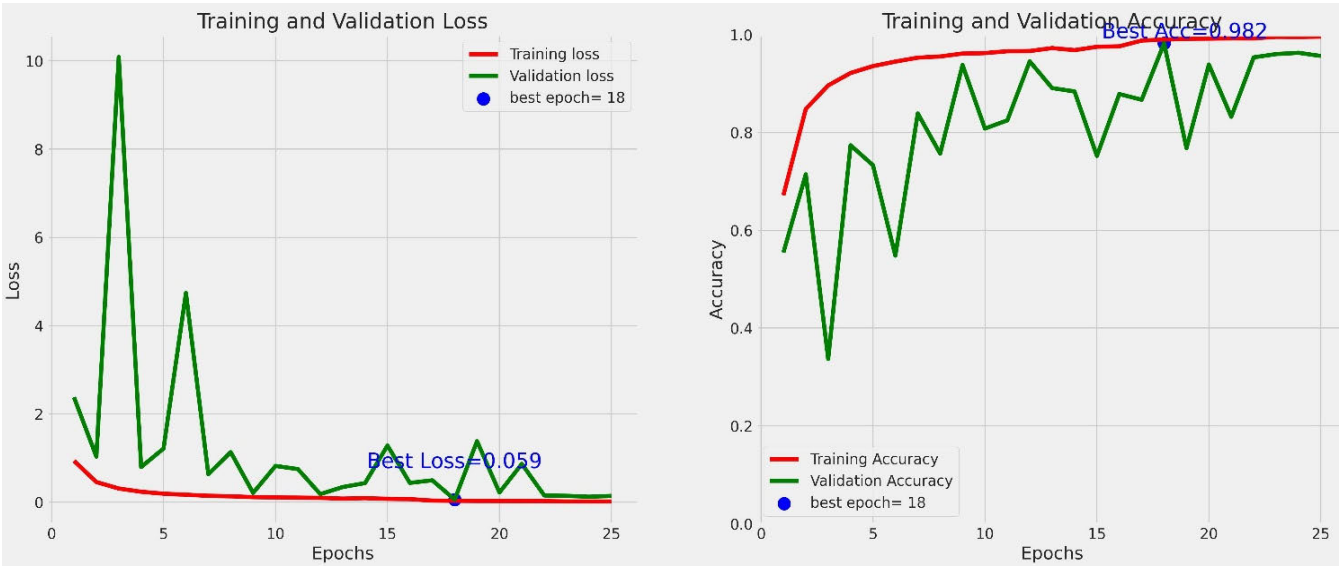

FIGURE 5. Training vs. validation accuracy and loss curves of experiment 1.

TABLE 1. Hyper parameters for FL-ToLeD model.

Hyper Parameters	Description
No. of convolution layers	4
No. of Separable convolution layers	2
No. of max pooling layers	6
Initial Learning rate	0.001
Max no. of epochs (Early Stopping)	50
Dropout rate	0.15
Batch Size	32
optimizer	Adam
Activation function	Relu & softmax
Loss function	categorical_crossentropy

TABLE 2. Dataset distribution (Train, Validation and Test).

Tomato leaves Disease	Trained Images	Validation Images	Tested images
Bacterial spot	1702	436	209
Early blight	1920	500	242
Healthy	1926	481	231
Late blight	1851	463	220
Leaf Mold	1882	470	242
Mosaic virus	1800	448	209
Septoria leaf spot	1745	436	220
Target Spot	1827	457	220
Two-spotted spider mite	1741	435	143
Yellow Leaf Curl Virus	1961	490	220
Total Leaves	18355	4616	2156

class over the minority classes biasing them towards the majority class. As a result, the model's ability to generalize and predict underrepresented groups is reduced, leading to poor training and evaluation performance. Another source of bias in PlantVillage is noise because it has noise associated with labels and deep learning models, we can easily exploit this bias to make prediction [34]. FL-ToLeD addresses these problems as we will see in the next sections.

B. EXPERIMENTS

FL-ToLeD was trained under a Python environment with TensorFlow, Keras, and other necessary libraries in Jupyter-Lab under Windows 10, 64-bit operating system. FL-ToLeD was trained with an initial learning rate of 0.001 for at most 50 epochs with early stopping to control overfitting problem and improve generalization of neural networks; other hyper-parameters are described in Table 1. These hyperparameters are selected manually and by using gride search algorithm [35].

Our initial dataset contains 25,127 images of tomato leaves that were distributed as described in Table 2. Next, we merged them all together into one list with their corresponding label, shuffling them randomly with state_random = 42 to avoid any bias that could occur in the classes of our dataset, we finally splitted them using two methods. In the first method, we split it 75:15:10 between the training, validation and testing dataset. The second method, we split using a 5-fold stratified cross-validation strategy of 80% for the training dataset with a validation split of 0.20 and 20% for testing the dataset. Two experiments were performed using 25,127 of tomato images.

1) EXPERIMENT 1

In this experiment, FL-ToLeD was fitted after splitting our images using the traditional split approach. With 18,845 training images, 5,653 validation images, and 629 test

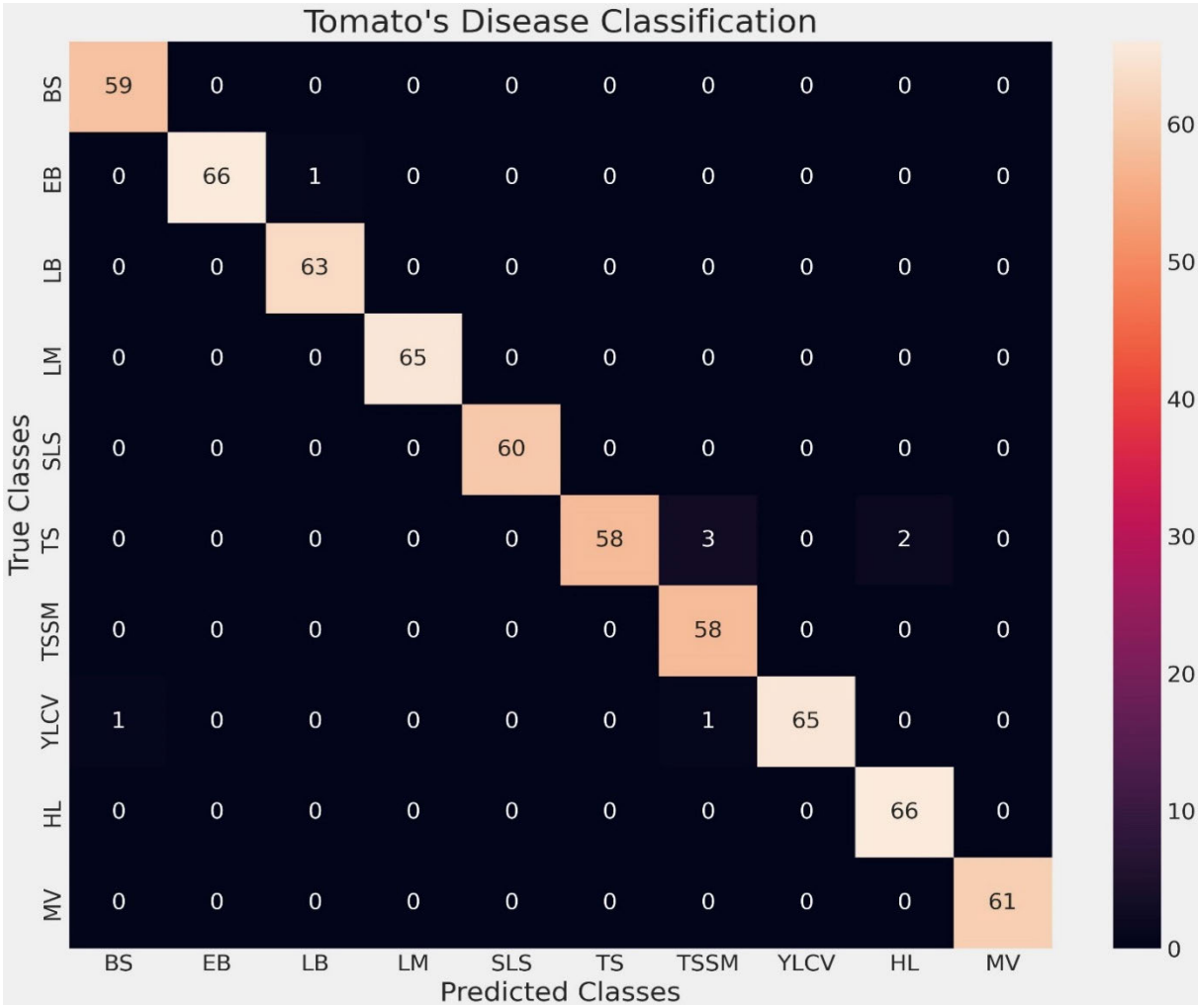

FIGURE 6. Confusion matrix of experiment 1.

images, the dataset was split in this proportion: 75% for training, 15% for validation, and 10% for testing. After we finished training and testing FL-ToLeD based on these hyper-parameters mentioned in Table 1, we observed that FL-ToLeD achieved 0.982, 0.059, 0.9872, 0.99, 0.99, 0.99 for validation accuracy, validation loss, testing accuracy, precision, recall and f1-score respectively with a model size of 2.5 MB and 221.594 of trainable parameters. Moreover, it takes about 3056.66 seconds to train and about 1.4 seconds in inference time. Training vs. validation accuracy and loss curves are shown in Figure 5. The best classification performance of FL-ToLeD is visualized using confusion matrix as shown in Figure 6. The worst validation accuracy occurred was 0.20 on epoch no. 3.

2) EXPERIMENT 2

In this experiment, we used a 5-fold stratified cross-validation strategy to split our images to fit FL-ToLeD. The dataset was divided as follows: 80% for training with a validation split of 0.20, 20% for testing, with 20,101 training images and 5,026 test images. After training and testing FL-ToLeD with the hyper-parameters listed in Table 1, we found that the best performance and complexity on the fold no. 1 with 0.995, 0.991, 0.028, 0.9904, 0.99, 0.99, 0.99 for training accuracy (TRA), validation accuracy (VA), validation loss (VL), testing accuracy (TA), precision (PRE), recall (REC) and f1-score (F1S) respectively with a model size of 2.5 MB and 221,594 of trainable parameters. Additionally, the training time (TRT) is 4209.65 seconds, and the inference time or test time (TET) is 10.40 seconds.

FL-ToLeD used ReduceLROnPlateau callback to reduce learning rate by a factor of 0.3 when a validation loss has stopped improving. This callback monitors the validation loss and if no improvement is seen for a 'patience=4', number of epochs, the learning rate is reduced automatically.

The best learning rate at which FL-ToLeD achieved its best accuracy on fold no. 1, epoch no. 34 is 0.00009 as shown in Figure 7.

The results for the remaining folds are shown in Table 3. The training versus validation accuracy and loss curves

TABLE 3. The results of 5-fold cross validation approach.

Fold No.	TRA	VA	VL	TA	TRT (Sec)	TET(Sec)	PRE %	REC %	F1S %
1	0.995	0.991	0.028	0.9904	4209.6595	10.4013	0.99	0.99	0.99
2	0.963	0.887	0.37	0.89196	1581.9225	10.31193	0.91	0.89	0.89
3	0.989	0.96	0.114	0.9588	3276.18589	10.18751	0.96	0.96	0.96
4	0.995	0.976	0.055	0.97592	4856.3258	10.34019	0.98	0.98	0.98
5	0.981	0.981	0.058	0.97711	2477.4366	10.2111	0.98	0.98	0.98

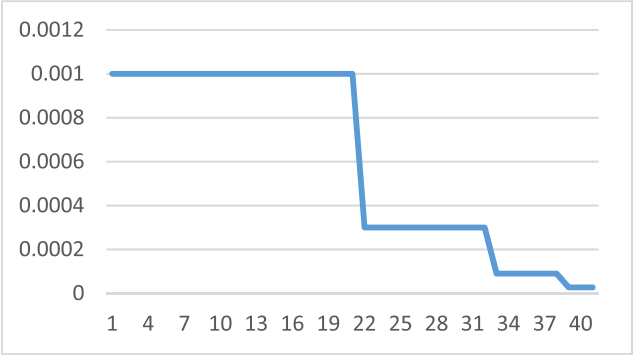

FIGURE 7. Learning rate per epochs.

for fold1, fold2, fold3, fold4 and fold5 are shown in Figure 8, Figure 9, Figure 10, Figure 11 and Figure 12 respectively.

C. PERFORMANCE ANALYSIS

In this section, we display the best classification performance of FL-ToLeD under a 5-fold stratified cross validation split. Table 4 displays precision, recall and f1-score for each individual class. Figure 13 displays the confusion matrix of the best model performance which has been achieved in fold no. 1. Figure 14 displays the AUC-ROC Curve for each of the classes with a weighted ROC score of 0.9998.

Finally, we applied a comparative study between our proposed model and seventeen of (CNN and Transfer learning) models in the terms of the used model architecture (MARCH), number of used images (NOI), image size (IS), dataset distribution (DIS), test accuracy (ACC), F1-Score (F1S), Precision (PRE), Recall (REC), no. of trainable parameters (NOTP) per million, model size (MS) per megabytes, Training time (TRT) per minutes as shown in Table 5. Our model is the best among the other models listed in Table 5, as it is the only one that achieves the trade-off between performance and computational complexity of the model, and this is the unique strength and main superiority of FL-ToLeD. Figure 15 also shows the superiority of FL-ToLeD over four CNN models.

TABLE 4. Report of performance on each class after evaluating FL-ToLeD.

Classes	Precision	Recall	F1-Score	Support
BS	0.98	0.98	0.98	469
EB	0.98	0.99	0.99	532
LB	0.99	0.98	0.99	506
LM	1.00	0.99	0.99	519
SLS	0.99	1.00	0.99	480
TS	0.98	0.99	0.98	501
TSSM	1.00	0.98	0.99	464
YLCV	0.99	0.99	0.99	535
HL	0.99	1.00	1.00	528
MV	1.00	1.00	1.00	492
accuracy			**0.99**	5026
macro avg	0.99	0.99	0.99	5026
weighted avg	0.99	0.99	0.99	5026

All previous comparison studies used the Plant Village dataset, the number of tomato plant disease images in the Plant Village dataset is equal to 18160 images, but in our study, we expanded the dataset using data augmentation technique to improve the model performance so that the number of images used is 25127. Furthermore, we split our dataset in two ways (75:15:10 ratio and 5 SK-Fold) using a 224-pixel image which varies from study to study. All these differences (NOI, IS, DIS) in the input dataset from one study to another affect the performance of the model. On the other hand, we divided the comparison into two groups, the first containing six transfer learning studies and the second containing eleven CNN studies. The potential strengths of our model as shown in the previous table are; it is the most lightweight model with **2.5 MB** with the best performance with **99.04%**, it is the only model between these compared models which proves the robustness and the generalizability of the model as we will discuss later. Individual contributions of each of the main components are also discussed in the next section.

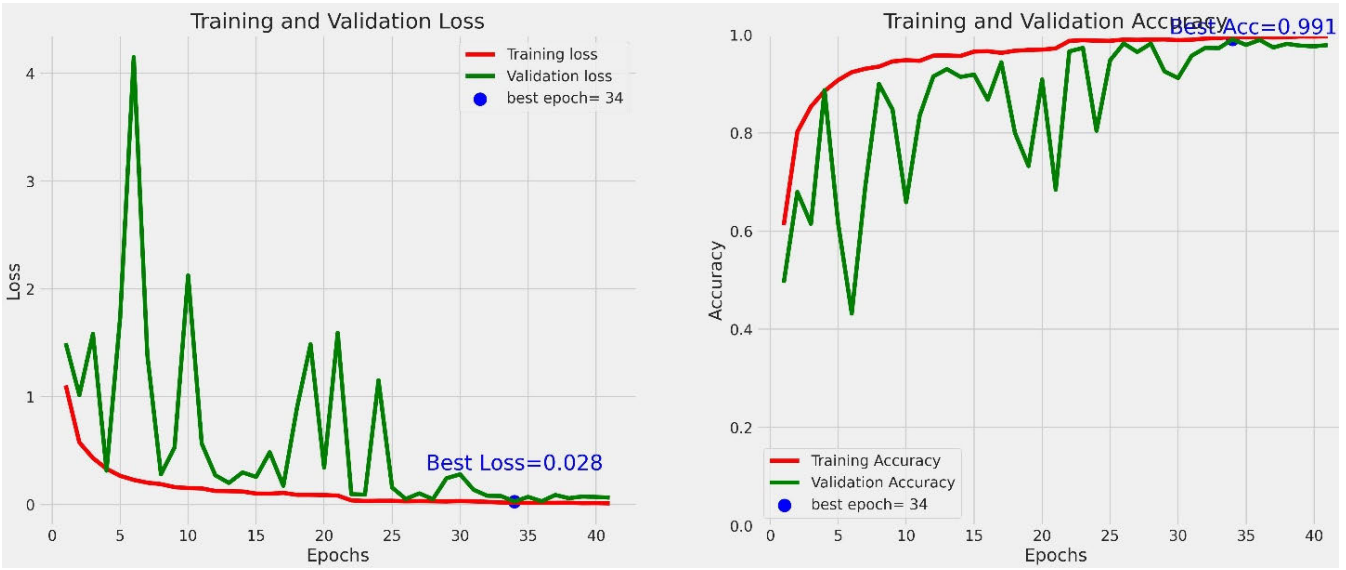

FIGURE 8. Training & validation (Loss & Accuracy) for Fold 1.

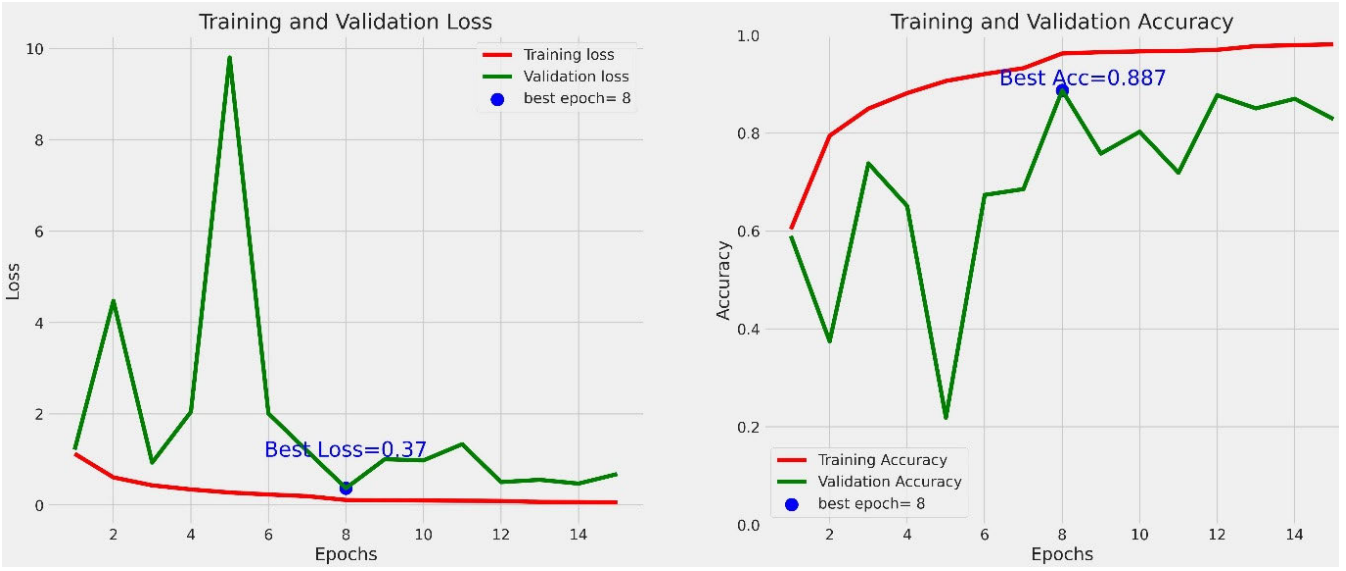

FIGURE 9. Training & validation (Loss & Accuracy) for Fold 2.

D. ABLATION STUDY

In this section, we analyze the impact or contribution of both depth-wise separable convolutional layers and soft attention-based mechanism on FL-ToLeD performance.

The depth-separable convolution layers (block no. 3) have a significant impact on the performance of FL-ToLeD such that by removing them from FL-ToLeD, we observed the best performance achieved on fold no. 1, epoch no. 37 with 0.9723, 0.972, and 0.08 for test accuracy, validation accuracy, and validation loss respectively as shown in Figure 16.

This proves the validity of what we discussed previously in section III-B about the extent of influence of depth-wise separable convolution layers with batch normalization layers on our model which has a high significance in reducing computational complexity and increasing the performance of the model.

Also, the soft attention block in FL-ToLeD has a great impact on FL-ToLeD performance as by removing it from the model, we observed that the best performance was achieved on fold no. 1, epoch no. 23 with 0. 9727, 0.981, 0.063 for test accuracy, validation accuracy, validation loss respectively as shown in Figure 17. This also proves the validity of what we discussed previously in section III-C about the extent of influence of soft attention operation on our model which has a high significance in increasing the model performance through highlight or paying attention to a specific area

FIGURE 10. Training & validation (Loss & Accuracy) for Fold 3.

FIGURE 11. Training & validation (Loss & Accuracy) for Fold 4.

of interest, increasing the value of relevant features and suppressing distracting features.

Moreover, by removing each of the depth-separable convolution layers (block no. 3) and soft attention block from FL-ToLeD, we observed the best performance achieved on fold no. 1, epoch no. 12 with 0.8927, 0.893, and 0.324 for test accuracy, validation accuracy, and validation loss respectively as shown in Figure 18. The total results are shown in Table 6.

E. QUALITATIVE ANALYSIS

The trade-off between the True Positive Rate and False Positive Rate by using various probability thresholds is described by the AUC-ROC curve, which may be used to evaluate the performance of a prediction model [46]. When the graph reaches 100% true positives and 0 false positives, the classifier is considered to be flawless. In general, we count the positive classifications by increasing the rate of false positives. We observed that FL-ToLeD performed admirably for all 10 classes since the AUC-ROC curves intersect each other in the upper left corner, as shown in Figure 13. This indicates that our proposed model classification performance is flawless in which it can accurately identify between all Positive and Negative class points with 0.998 weighted ROC score.

To qualitatively evaluate and analyze our model and explain why our model predicts what it predicts, we use Grad-CAM.

Grad-CAM which stands for Gradient-weighted Class Activation Mapping [47] is a localization of model decisions

FIGURE 12. Training & validation (Loss & Accuracy) for Fold 5.

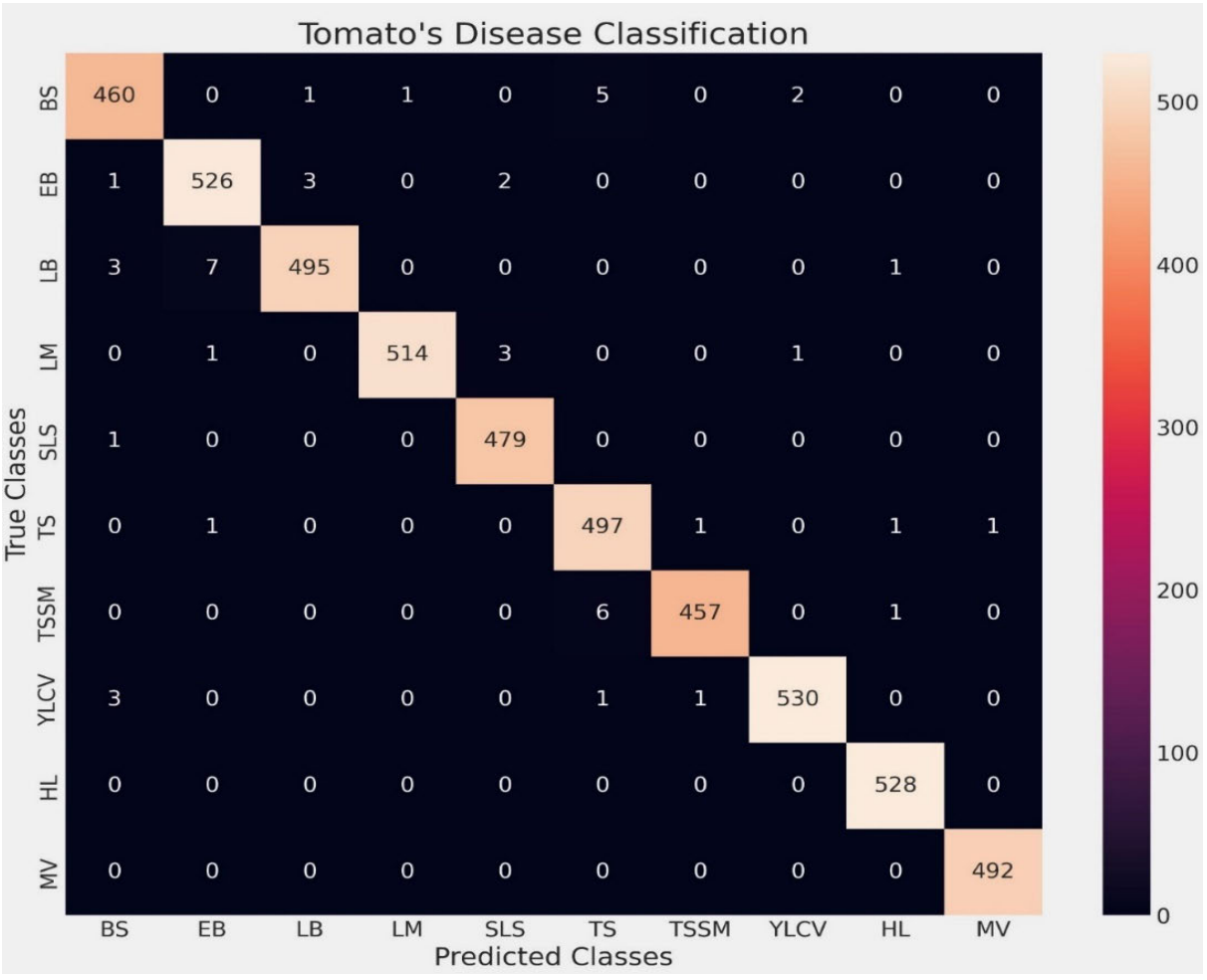

FIGURE 13. Confusion matrix of the best model performance occurred in Fold 1.

approach. It creates a coarse localization map that highlights the key areas in the image for concept prediction by using gradients of any target concept, this localization map helps understand the specific features or areas in an image that

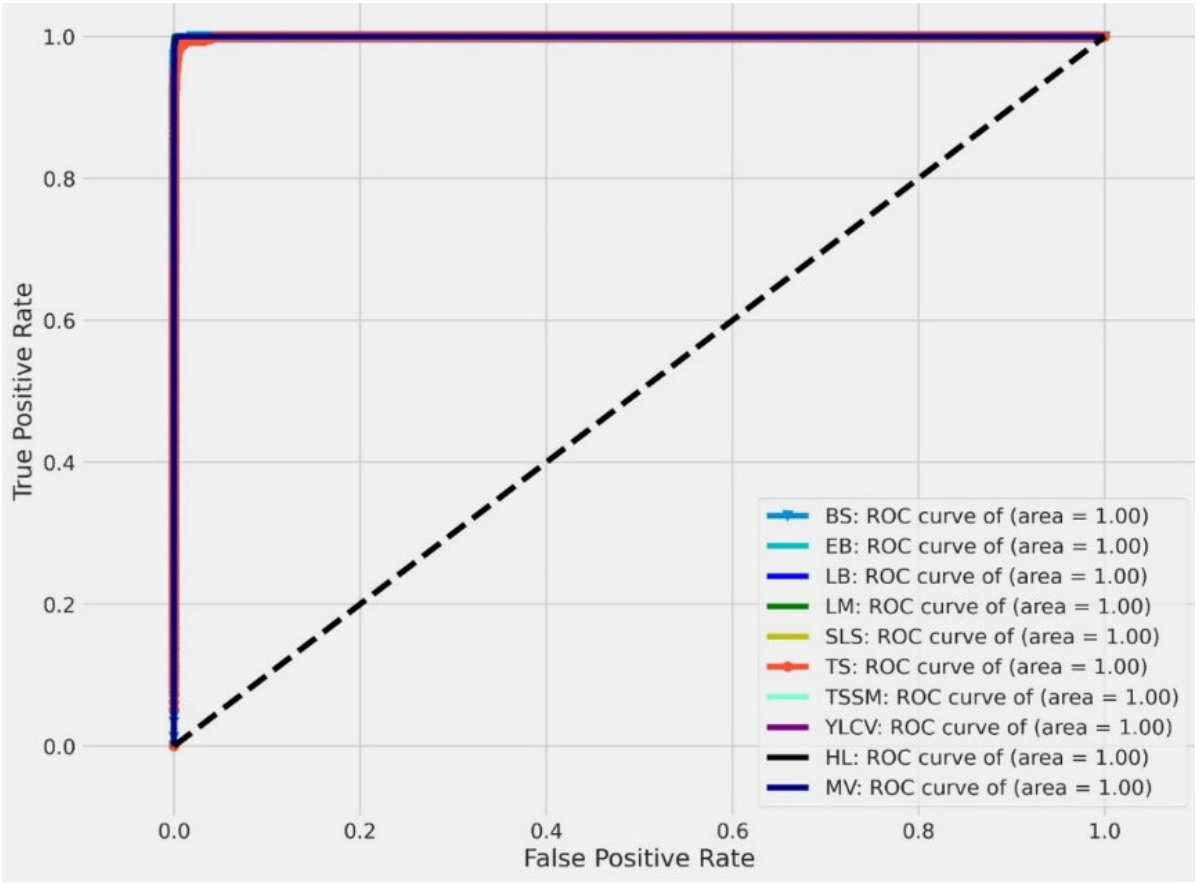

FIGURE 14. AUC-ROC curve for each of class.

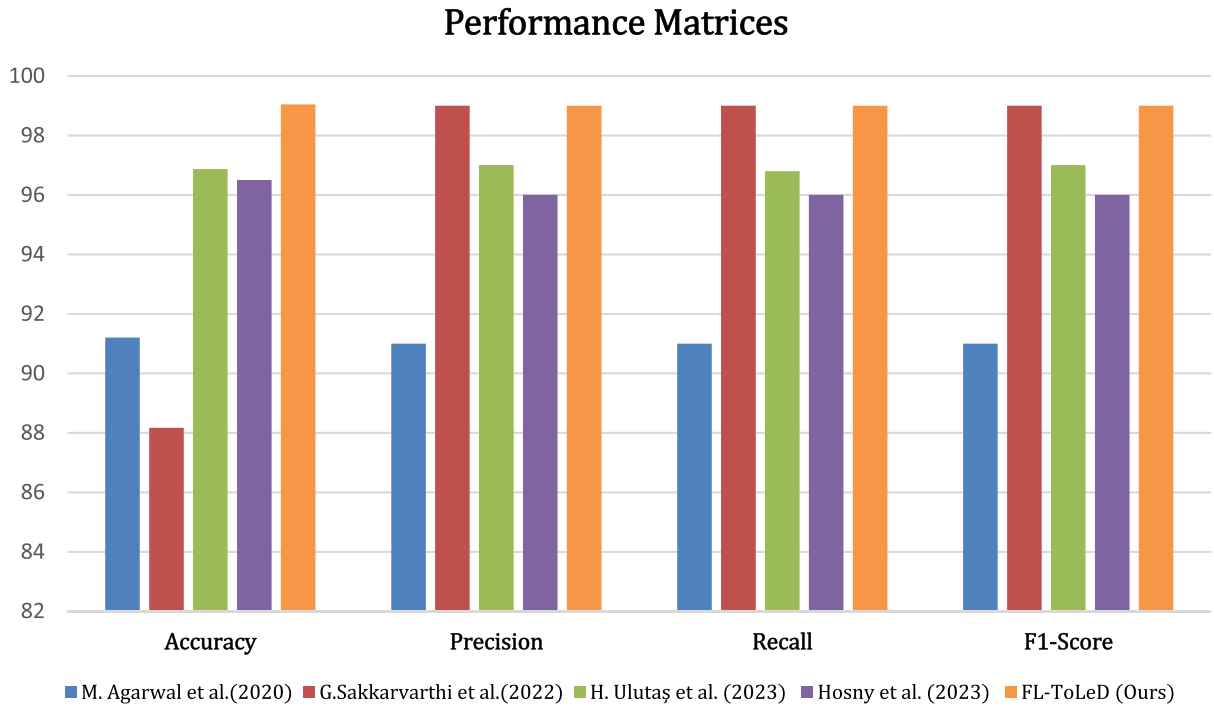

FIGURE 15. Comparison classification performance of our proposed FL-ToLeD with other CNN models.

TABLE 5. Comparative study of FL-ToLeD with existing tomato diseases classification models.

Ref. No.	MARCH	Database			Model Performance (%)				Model Complexity		
		NOI	IS	DIS	ACC	PRE	REC	F1S	NOTP	MS	TRT
[9]	MobileNetV2	18160	256	60:20:20	99.3	99.18	99.07	99.12	–	9.60	–
[15]	AlexNet	1909	227	80:20	93.4	92.99	93.02	93	61	227	–
[10]	DenseNet121 with C-GAN	16012	224	60:10:30	97.11	97	97	97	1.7	–	–
[36]	EfficientNetB5	11000	200	10 K-Fold	99.07	99.5	99.5	99.5	28.8	–	–
[37]	VGG19 with Inception-V3	18160	256	60:30:10	94.25	90.08	63.82	93.89	–	–	–
[38]	VGGNet with two inception blocks	18160	224	70:20:10	99.2	99.1	99.2	99.2	21	84	82
[39]	CNN	3000	128	70:30	88.17	99	99	99	1.1	–	–
[40]	CNN	18160	224	5 K-Fold	96.87	97	96.8	97	0.494	6	82
[41]	DCNN with LBP	18160	64	80:20	96.5	96	96	96	0.405	–	–
[42]	CNN	12693	150	90:10	97.36	97	97	97	9.5	108	114
[43]	Hybrid CNN	18160	224	90:10	99.17	99.13	99.23	99.17	–	–	–
[44]	CNN	16012	150	80:20	98.19	–	–	–	3	–	–
[19]	CNN with self-attention	19510	256	80:10:10	99.69	99.69	99.69	99.69	1.44	16.8	225
[11]	CNN	3000	256	80:20	98.49	98	98	98	1.42	–	–
[45]	CNN with attention mechanism	22925	224	80:20	96.81	96.77	96.81	96.79	–	–	–
[14]	CNN	17500	224	57:40:3	91.2	91	91	91	0.2	1.5	–
[16]	CNN with Attention Residual	95999	256	5 K-Fold	98.0	–	–	–	0.6	–	600
Ours	**CNN with depth-wise separable Convolution & soft attention**	**25127**	**224**	**5 SK-Fold**	**99.04**	**99**	**99**	**99**	**0.221**	**2.5**	**70**
		25127	**224**	**75:15:10**	**98.72**	**99**	**99**	**99**	**0.221**	**2.5**	**51**

the model focuses on when making predictions and provides visual explanations and helps understand how CNNs arrive at their predictions.

Grad-CAM uses the spatial data (feature map) produced in the last convolutional layer to produce the localization map to identify salient regions of tomato images that were useful in the classification process.

The last convolutional layer in FL-ToLeD which it used to generate the heatmap is called ''separable_conv2d_2''. We observed that FL-ToLeD focuses more on the relevant regions of tomato leaves that have the greatest impact on classification. Figure 19 displays FL-ToLeD 's performance on unseen random tested images from the tested dataset with the prediction heatmap for the most important features or regions for each prediction in which ''TLAB'' stands for True Label or class and ''PLAB'' stands for Predicted Label or class.

Generalizability is one of the most challenging issues when it comes to model performance. It mainly comes down to how well the model performs on unseen data. There are many

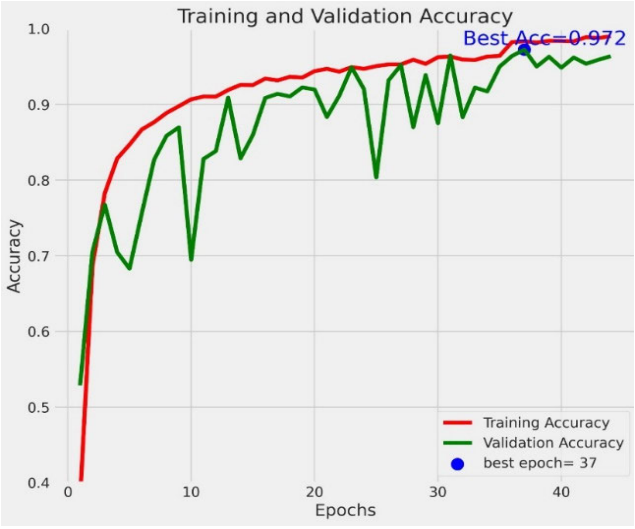

FIGURE 16. Model performance without depth-separable convolution layers.

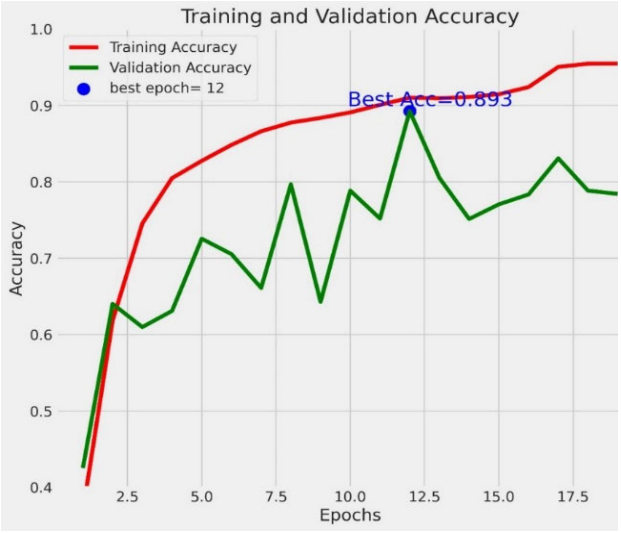

FIGURE 18. Model performance without both of depth-separable convolution layers and soft attention-based mechanism block.

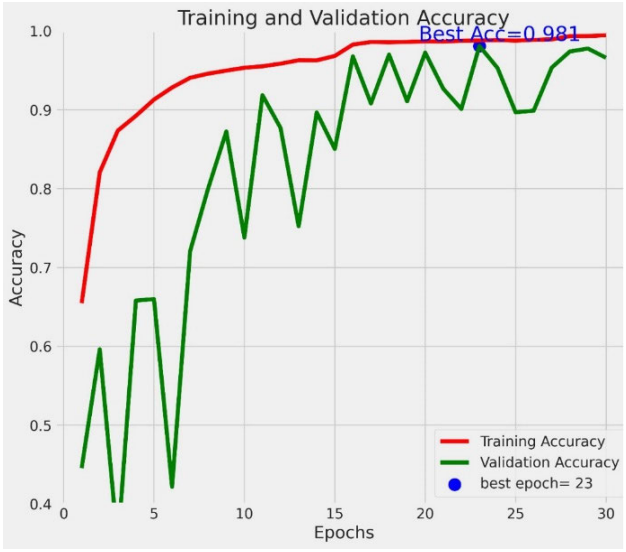

FIGURE 17. Model performance without soft attention-based mechanism.

TABLE 6. Impact of depth-separable convolution layers and soft attention blocks on FL-ToLeD performance.

	Without		
	Block 3	Attention Block	Both Blocks
# of Trainable Params	103,770	182,154	93,898
Train time (Sec)	4286.58	2705.49	1695.23
Prediction time (Sec)	9.956	9.675	9.71
Validation loss	0.080	0.063	0.324
Validation Accuracy (%)	**97.20**	**98.10**	**89.30**
Test Accuracy (%)	97.23	97.27	89.27
Precision (%)	97.00	97.00	90.00
Recall (%)	97.00	97.00	89.00
F1-score (%)	97.00	97.00	89.00

ways to increase the generalizability of CNN models [48] but there are two most common solutions: data augmentation and data normalization and FL-ToLeD used both as we discussed later.

Another technique used to standardize the inputs to a layer for each mini-batch is called batch normalization as we discussed later. It not only provides regularization and reduces generalization error, but also speeds up training and reduces model complexity.

F. QUANTITATIVE ANALYSIS

When we tested the model with different splits on the plantvillage dataset, we discovered that the model using five stratified cross validation outperforms the model using traditional split (75%, 15%, and 10%) by 0.5%. Hence, we selected five stratified cross validation split for running our experiments as we test FL-ToLeD on 5 different unseen folds.

Furthermore, the proposed model is compared with state-of-the-art models for tomato leaves diseases classification on the plantvillage dataset. We discovered that our depth-wise separable convolutions with the soft Attention based mechanism outperforms H. Ulutaş et al. [20] by 2.2% in the terms of accuracy, precision, recall and f1-score. It also outperformed Khamparia et al. [13] by 8% in the terms of accuracy, precision, recall and f1-score.

Furthermore, from the ablation study, we observed that the proposed model has lost about 10% of the accuracy by removing depth-wise separable convolutions layers and soft Attention based mechanism layers together from the base model as it achieved about 0.8927% of test accuracy. From the aspect of model complexity, we observed that

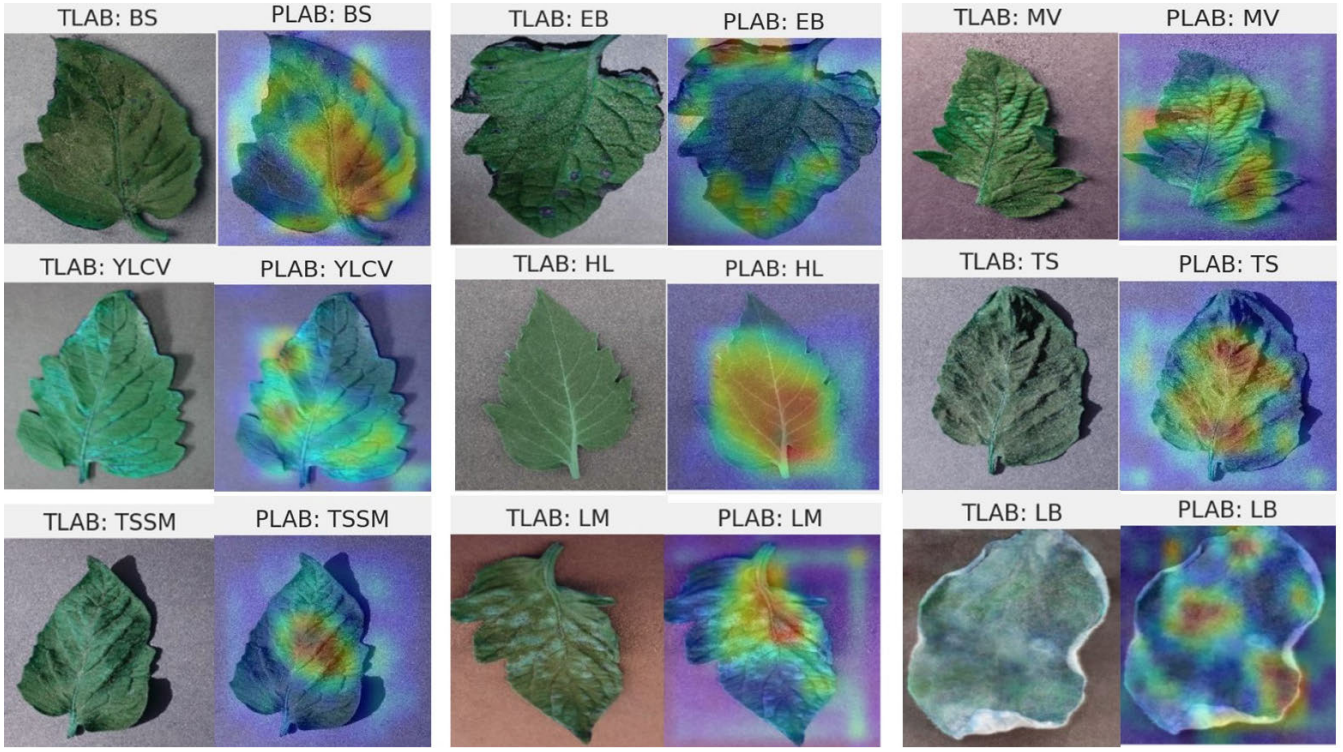

FIGURE 19. The visual explanations of the relevant regions of the leaves that have the greatest influence on classification using Grad-Cam.

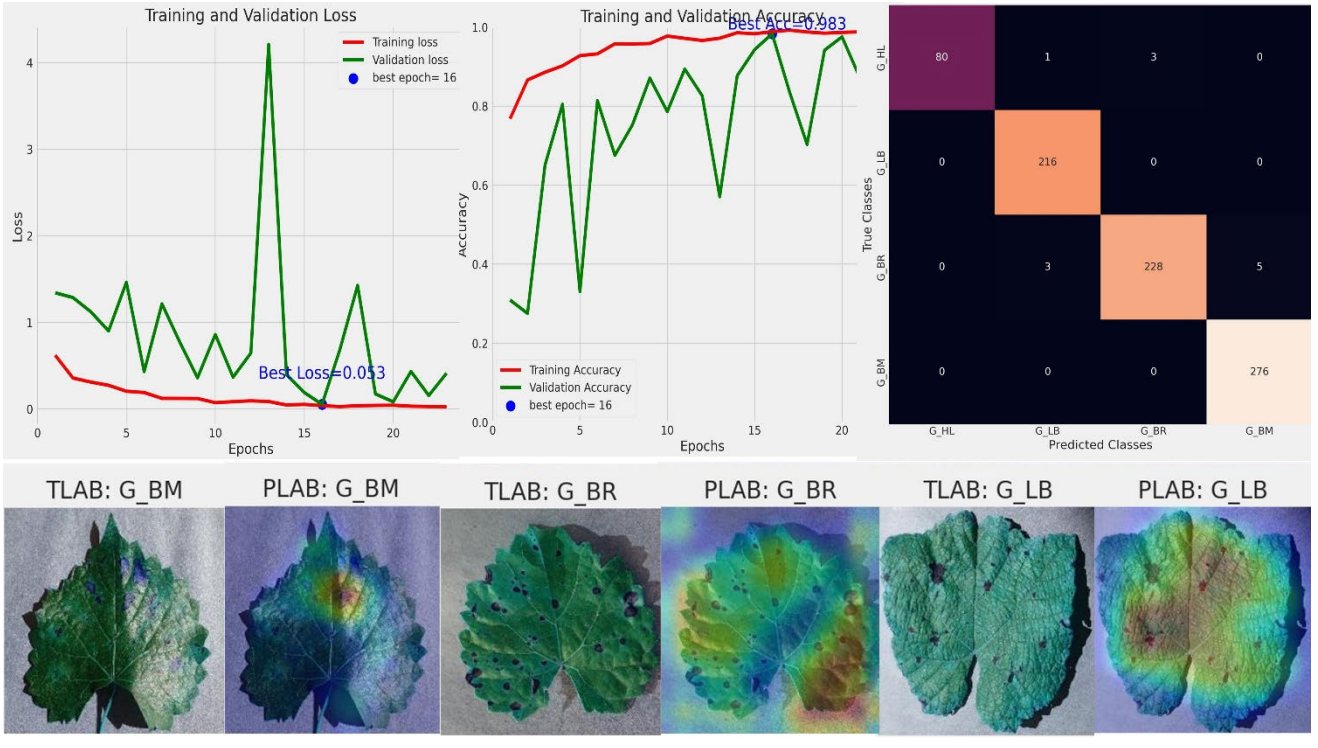

FIGURE 20. The performance of FL-ToLeD on Grape diseases leaves images dataset (Confusion matrix, grad-cam, accuracy curves and loss curves).

there is only one model that outperformed FL-ToLeD as Khamparia et al. [13] built their model with size=1.5 MB but achieve non satisfactory performance as it achieved about 91% performance, this validates what we discussed earlier in Section III-B and Section III-C about how depth-separable convolution layers combined with batch normalization layers

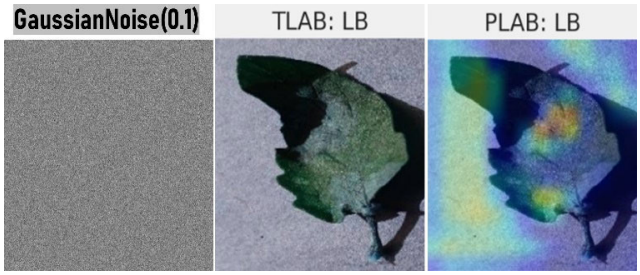

FIGURE 21. Effect of using Gaussian noise on the robustness of FL-ToLeD.

and soft attention operation impact the complexity and performance of our model.

Experimental results show that FL-ToLeD works more accurately and efficiently than the comparative models presented in Table 4 as FL-ToLeD achieved an accuracy of 99.04% with a model size of 2.5 MB.

Moreover, in order to verify the practical application performance of the FL-ToLeD, we conducted experiments on grape leaf diseases. The dataset contains 4062 images of grape leaf diseases including 3 diseases; Black_Measles'' BM'', Black_rot ''BR'', Leaf_blight ''LB'' and one healthy class. After training and testing FL-ToLeD with the hyper-parameters listed in Table 1, we found that the best performance as shown in Figure 20 was on fold no. 4 with 0983, 0.053, **0.9852**, 0.99, 0.98, 0.98 for validation accuracy, validation loss, test accuracy, precision, recall, f1-score respectively.

Model's robustness [49], [50] refers to the ability of a model to maintain its performance when faced with uncertainties or adverse conditions such as dealing with noisy data, shifted or modified data. Evaluating and measuring the robustness of deep learning models is certainly one of the most important research problems as it can help us better understand the weaknesses of them and also form a basis for designing a more robust neural network in the future.

In real-world applications, image quality may vary widely depending on factors such as the capture sensor used and lighting conditions. These scenarios should be taken into account when performing image classification, as changing the quality directly affects its results so that the robustness must be certified before a neural network is deployed in real-world applications.

The loss function plays an important role in the noise robustness of CNN models. Loss sensitivity represents the loss function robustness for noisy data. Here, we use the cross-entropy loss function as it is the most robust loss function to noises [51]. Some of the most common types of noise that we can add to the images are; Poisson noise, Salt and pepper noise, Gaussian noise and others. To test the robustness of FL-ToLeD, we add a Gaussian noise to the tested images with standard deviation=0.1 of the noise distribution as shown in Figure 21. We observed that, the performance of FL-ToLeD didn't suffer so much as it achieved about **0.98925** of test accuracy with this input noisy images. Finally, FL-ToLeD has the ability to maintain its performance when faced with noise.

V. CONCLUSION

In this paper, we developed a lightweight convolutional neural network model based on an attention-based mechanism and deep separable convolutional layers that outperformed the majority of existing models of tomato plant disease in terms of model performance and computational complexity. This model achieved an accuracy of 99.04% with a model size of 2.5 MB, which makes it better suited for embedding on low-end devices and meeting real-time requirements. The proposed model offers a good compromise between effectiveness and performance.

As a result, it can be used in mobile applications and other embedded devices with limited resources, helping spread awareness of precision farming applications. FL-ToLeD used a runtime data augmentation method, which improved the generalizability of the network and protected it from the problem of overfitting. In addition, the results of our experiment show the superiority of the proposed method over those currently used.

The benefits of FL-ToLeD are:

- It can be used to detect tomato plant diseases in the early stages, as small lesions can be identified in the leaves.
- It can be included on any low-end device to use in real life scenarios since its size is very small (2.5MB).
- FL-ToLeD can identify the disease with a very high accuracy (99.04%), which is superior to others currently used.

The potential challenges and limitations of deploying FL-ToLeD in real-world precision agriculture settings are:

- Due to the presence of light shadow, density of overlapping leaves, obstruction of branches, and overlapping of leaves in the real natural environment, identification and localization of the disease under these conditions may be a major challenge.
- In different monitoring and imaging equipment, the distance between plants and imaging equipment is different, resulting in different resolution, light and perspective of the image of plant diseases, and also different optical characteristics of the same image of plant diseases and pests will produce visible changes.
- The varying degree of occlusion resulting from background and other factors leads to a large number of occlusion problems, which may lead to poor recognition of plant diseases.
- The main limitation or drawback of the model is that it has been tested only on leaves of plants but in future version it will deal with other parts of plants like stems.

Moreover, the potential future directions of our research are:

- We plan to upgrade FL-ToLeD in terms of data privacy, availability and transfer costs through federated learning approach.

- We plan to adapt our model to IoT applications.
- We would also like to improve FL-ToLeD so that it can report disease prevalence in plants.
- We will focus more on identifying diseases at different sites on plants and trees, such as fruits, flowers and stems.

REFERENCES

[1] S. M. El-Ganainy, "Morphological and molecular characterization of large-spored alternaria species associated with potato and tomato early blight in Egypt," *Int. J. Agricult. Biol.*, vol. 25, no. 5, pp. 1101–1110, May 2021.

[2] S. Panno, S. Davino, A. G. Caruso, S. Bertacca, A. Crnogorac, A. Mandić, E. Noris, and S. Matić, "A review of the most common and economically important diseases that undermine the cultivation of tomato crop in the Mediterranean basin," *Agronomy*, vol. 11, no. 11, p. 2188, Oct. 2021.

[3] R. Zhou, S. Kaneko, F. Tanaka, M. Kayamori, and M. Shimizu, "Disease detection of cercospora leaf spot in sugar beet by robust template matching," *Comput. Electron. Agricult.*, vol. 108, pp. 58–70, Oct. 2014.

[4] L. Liu, Y. Dong, W. Huang, X. Du, B. Ren, L. Huang, Q. Zheng, and H. Ma, "A disease index for efficiently detecting wheat fusarium head blight using Sentinel-2 multispectral imagery," *IEEE Access*, vol. 8, pp. 52181–52191, 2020.

[5] E. Moriones and J. Navas-Castillo, "Tomato yellow leaf curl virus, an emerging virus complex causing epidemics worldwide," *Virus Res.*, vol. 71, nos. 1–2, pp. 123–134, Nov. 2000.

[6] G. Idoje, T. Dagiuklas, and M. Iqbal, "Survey for smart farming technologies: Challenges and issues," *Comput. Electr. Eng.*, vol. 92, Jun. 2021, Art. no. 107104.

[7] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, pp. 1–74, Mar. 2021.

[8] M. Alajanbi, D. Malerba, and H. Liu, "Distributed reduced convolution neural networks," *Mesopotamian J. Big Data*, vol. 2021, pp. 26–29, Jul. 2021.

[9] S. Ahmed, Md. B. Hasan, T. Ahmed, Md. R. K. Sony, and Md. H. Kabir, "Less is more: Lighter and faster deep neural architecture for tomato leaf disease classification," *IEEE Access*, vol. 10, pp. 68868–68884, 2022.

[10] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with C-GAN synthetic images," *Comput. Electron. Agricult.*, vol. 187, Aug. 2021, Art. no. 106279.

[11] N. K. Trivedi, V. Gautam, A. Anand, H. M. Aljahdali, S. G. Villar, D. Anand, N. Goyal, and S. Kadry, "Early detection and classification of tomato leaf disease using high-performance deep neural network," *Sensors*, vol. 21, no. 23, p. 7987, Nov. 2021.

[12] A. A. Ali, B. Chramcov, R. Jasek, R. Katta, S. Krayem, and E. Awwama, "Tomato leaf diseases detection using deep learning," in *Computational Methods in Systems and Software*. Cham, Switzerland: Springer, 2021, pp. 199–208.

[13] A. Khamparia, G. Saini, D. Gupta, A. Khanna, S. Tiwari, and V. H. C. de Albuquerque, "Seasonal crops disease prediction and classification using deep convolutional encoder network," *Circuits, Syst., Signal Process.*, vol. 39, no. 2, pp. 818–836, Feb. 2020.

[14] M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, "ToLeD: Tomato leaf disease detection using convolution neural network," *Proc. Comput. Sci.*, vol. 167, pp. 293–301, Sep. 2020.

[15] S. Verma, A. Chug, and A. P. Singh, "Application of convolutional neural networks for evaluation of disease severity in tomato plant," *J. Discrete Math. Sci. Cryptography*, vol. 23, no. 1, pp. 273–282, Jan. 2020.

[16] R. Karthik, M. Hariharan, S. Anand, P. Mathikshara, A. Johnson, and R. Menaka, "Attention embedded residual CNN for disease detection in tomato leaves," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105933.

[17] A. Elhassouny and F. Smarandache, "Smart mobile application to recognize tomato leaf diseases using convolutional neural networks," in *Proc. Int. Conf. Comput. Sci. Renew. Energies (ICCSRE)*, Jul. 2019, pp. 1–4.

[18] J. Costa, C. Silva, and B. Ribeiro, "Hierarchical deep learning approach for plant disease detection," in *Proc. 9th Iberian Conf.*, 2019, pp. 383–393.

[19] A. Bhujel, N.-E. Kim, E. Arulmozhi, J. K. Basak, and H.-T. Kim, "A lightweight attention-based convolutional neural networks for tomato leaf disease classification," *Agriculture*, vol. 12, no. 2, p. 228, Feb. 2022.

[20] Y. Feng, M. Gao, and Z. Zhang, "Web service QoS classification based on optimized convolutional neural network," in *Proc. IEEE 14th Int. Conf. Intell. Syst. Knowl. Eng. (ISKE)*, Nov. 2019, pp. 584–590.

[21] M. Kaushik, P. Prakash, R. Ajay, and S. Veni, "Tomato leaf disease detection using convolutional neural network with data augmentation," in *Proc. 5th Int. Conf. Commun. Electron. Syst. (ICCES)*, Jun. 2020, pp. 1125–1132.

[22] S. Tuffery, "Deep learning," in *Deep Learning: From Big Data to Artificial Intelligence With R*. Cambridge, MA, USA: MIT Press, 2023, pp. 269–346.

[23] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015.

[24] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 1–18.

[25] V. Nair and G. E. Hinton., "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1–18.

[26] R. Shang, J. He, J. Wang, K. Xu, L. Jiao, and R. Stolkin, "Dense connection and depthwise separable convolution based CNN for polarimetric SAR image classification," *Knowledge-Based Syst.*, vol. 194, Apr. 2020, Art. no. 105542.

[27] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[28] R. H. Hridoy, T. Habib, I. Jabiullah, R. Rahman, and F. Ahmed, "Early recognition of betel leaf disease using deep learning with depth-wise separable convolutions," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, Aug. 2021, pp. 1–7.

[29] V. Thakkar, S. Tewary, and C. Chakraborty, "Batch normalization in convolutional neural networks—A comparative study with CIFAR-10 data," in *Proc. 5th Int. Conf. Emerg. Appl. Inf. Technol. (EAIT)*, Jan. 2018, pp. 1–5.

[30] N. Tomita, B. Abdollahi, J. Wei, B. Ren, A. Suriawinata, and S. Hassanpour, "Attention-based deep neural networks for detection of cancerous and precancerous esophagus tissue on histopathological slides," *JAMA Netw. Open*, vol. 2, no. 11, Nov. 2019, Art. no. e1914645.

[31] M. A. Shaikh, T. Duan, M. Chauhan, and S. N. Srihari, "Attention based writer independent verification," in *Proc. 17th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Sep. 2020, pp. 373–379.

[32] D. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," 2015, *arXiv:1511.08060*.

[33] A. T. Khan, S. M. Jensen, A. R. Khan, and S. Li, "Plant disease detection model for edge computing devices," *Frontiers Plant Sci.*, vol. 14, Dec. 2023, Art. no. Frontiers in Plant Science.

[34] M. Alican Noyan, "Uncovering bias in the PlantVillage dataset," 2022, *arXiv:2206.04374*.

[35] B. H. Shekar and G. Dagnew, "Grid search-based hyperparameter tuning and classification of microarray cancer data," in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Paradigms (ICACCP)*, Feb. 2019, pp. 1–8.

[36] M. Bhandari, T. B. Shahi, A. Neupane, and K. B. Walsh, "BotanicX-AI: Identification of tomato leaf diseases using an explanation-driven deep-learning model," *J. Imag.*, vol. 9, no. 2, p. 53, Feb. 2023.

[37] F. Arshad, M. Mateen, S. Hayat, M. Wardah, Z. Al-Huda, Y. H. Gu, and M. A. Al-antari, "PLDPNet: End-to-end hybrid deep learning framework for potato leaf disease prediction," *Alexandria Eng. J.*, vol. 78, pp. 406–418, Sep. 2023.

[38] T. Sanida, A. Sideris, M. V. Sanida, and M. Dasygenis, "Tomato leaf disease identification via two–stage transfer learning approach," *Smart Agricult. Technol.*, vol. 5, Oct. 2023, Art. no. 100275.

[39] G. Sakkarvarthi, G. W. Sathianesan, V. S. Murugan, A. J. Reddy, P. Jayagopal, and M. Elsisi, "Detection and classification of tomato crop disease using convolutional neural network," *Electronics*, vol. 11, no. 21, p. 3618, Nov. 2022.

[40] H. Ulutaş and V. Aslantaş, "Design of efficient methods for the detection of tomato leaf disease utilizing proposed ensemble CNN model," *Electronics*, vol. 12, no. 4, p. 827, Feb. 2023.

[41] K. M. Hosny, W. M. El-Hady, F. M. Samy, E. Vrochidou, and G. A. Papakostas, "Multi-class classification of plant leaf diseases using feature fusion of deep convolutional neural network and local binary pattern," *IEEE Access*, vol. 11, pp. 62307–62317, 2023.
[42] H. I. Peyal, M. Nahiduzzaman, M. A. H. Pramanik, M. K. Syfullah, S. M. Shahriar, A. Sultana, M. Ahsan, J. Haider, A. Khandakar, and M. E. H. Chowdhury, "Plant disease classifier: Detection of dual-crop diseases using lightweight 2D CNN architecture," *IEEE Access*, vol. 11, pp. 110627–110643, 2023.
[43] M. V. Sanida, T. Sanida, A. Sideris, and M. Dasygenis, "An efficient hybrid CNN classification model for tomato crop disease," *Technologies*, vol. 11, no. 1, p. 10, Jan. 2023.
[44] P. Baser, J. R. Saini, and K. Kotecha, "TomConv: An improved CNN model for diagnosis of diseases in tomato plant leaves," in *Proc. Comput. Sci.*, vol. 218, Jan. 2023, pp. 1825–1833.
[45] S. Zhao, Y. Peng, J. Liu, and S. Wu, "Tomato leaf disease diagnosis based on improved convolution neural network by attention module," *Agriculture*, vol. 11, no. 7, p. 651, Jul. 2021.
[46] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Mach. Learn.*, vol. 45, no. 2, pp. 171–186, Nov. 2001.
[47] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.
[48] C. Chen, W. Bai, R. H. Davies, A. N. Bhuva, C. H. Manisty, J. B. Augusto, J. C. Moon, N. Aung, A. M. Lee, M. M. Sanghvi, K. Fung, J. M. Paiva, S. E. Petersen, E. Lukaschuk, S. K. Piechnik, S. Neubauer, and D. Rueckert, "Improving the generalizability of convolutional neural network-based segmentation on CMR images," *Frontiers Cardiovascular Med.*, vol. 7, p. 105, Jun. 2020.
[49] P. Arcaini, A. Bombarda, S. Bonfanti, and A. Gargantini, "Dealing with robustness of convolutional neural networks for image classification," in *Proc. IEEE Int. Conf. Artif. Intell. Test.*, Aug. 2020, pp. 7–14.
[50] S. Ge, Z. Xia, J. Fei, Y. Tong, J. Weng, and M. Li, "A robust document image watermarking scheme using deep neural network," *Multimedia Tools Appl.*, vol. 82, no. 25, pp. 38589–38612, Oct. 2023.
[51] J. Lv, B. Liu, L. Feng, N. Xu, M. Xu, B. An, G. Niu, X. Geng, and M. Sugiyama, "On the robustness of average losses for partial-label learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 1–15, May 2024.

MAHMOUD H. ALNAMOLY received the bachelor's and M.Sc. degrees from the Department of Computer Science, Faculty of Computer and Informatics, Zagazig University, Egypt, in 2015 and 2020, respectively. He is currently a Teaching Assistant with the Department of Computer Science, Faculty of Computer and Informatics, Zagazig University. His research interests include artificial intelligence algorithms, computer science applications, bioinformatics, natural language processing, and computer vision.

ANAR A. HADY (Member, IEEE) received the B.Eng. degree from the Computer Engineering and Systems Department, Faculty of Engineering, Ain Shams University, in 2002, the M.Sc. degree from the Faculty of Engineering, Cairo University, in 2007, and the Ph.D. degree from the Faculty of Engineering, Ain Shams University, in 2014. From 2018 to 2019, she was a Postdoctoral Scholar of computer science and engineering with Washington University in St Louis, Missouri, MO, USA. She is currently an Associate Professor with the Electronics Research Institute (ERI), Egypt. Since 2022, she has been an Associate Professor with The British University in Egypt (BUE). She is the author of many articles in reputable journals, conferences, and book chapters. She has a copyright and one patent in precision agriculture. She was a Co-PI of an accomplished project for developing a prototype of a sensor network for precision agriculture and is currently a PI for a cloud-sensing project. Her research interests include wireless sensor networks, network security, and the Internet of Things.

SHERINE M. ABD EL-KADER received the M.Sc. degree from the Electronics and Communications Department, Faculty of Engineering, Cairo University, in 1998, and the Ph.D. degree from the Computers Engineering Department, Faculty of Engineering, Cairo University, in 2003. She was the Head of the Internet and Networking Unit, Electronics Research Institute (ERI), from 2003 to 2014; the Information and Decision Making Support Center, ERI, from 2009 to 2014; an Associate Professor with the Faculty of Engineering, Akhbar El Yom Academy, from 2007 to 2009; and the Head of the Technology Innovation Support Center (TISC), ERI, from 2013 to 2017, where she has been a Professor with the Computers and Systems Department, since April 2014. She has been a member of the Technological Development and Linkage with the Industry and Civil Society Committee, since 2016, and has been the Head of the Media Committee, since 2017. She has been the Head of the Computers and Systems Department, ERI, since 2018. She has also been the Head of the Technology, Innovation and Commercialization Office (TICO), since 2018. She is currently the President of ERI. She has supervised more than 20 M.Sc. and Ph.D. students. She has published more than 50 papers and six book chapters in the computer networking area, and an editor of a book on 5G and precision agriculture. She is working on many computer networking hot topics, such as the IoT, 5G, cognitive radio, Wi-MAX, Wi-Fi, IP mobility, QoS, wireless sensor networks, ad-hoc networking, real-time traffic, and localization algorithms. She has supervised many automation and web projects for ERI. She was also a Technical Member of the ERI Projects Committee and the Telecommunication Networks Committee, Egyptian Organization for Standardization and Quality, from 2007 to 2011. She has five copyrights and one patent with a great background in intellectual property. She is also a technical reviewer for many international journals.

IBRAHIM EL-HENAWY was the Head of the Department of Computer Science, Faculty of Computer and Informatics, Zagazig University, Egypt, where he is currently a Professor of computer science. His research interests include computer science applications, mathematics, artificial intelligence techniques, computer vision, digital image processing, soft computing, and optimization.